

***** Computer Select, December 1996 : Articles *****

Journal: DBMS July 1996 v9 n8 p20(4)
COPYRIGHT M&T Publishing Inc. 1996

Title: Soothsayers for Oracle.
(Java, Oracle tuning products and a puzzle) (SQL for
Smarties)
Author: Celko, Joe

Full Text:

JOE ON JAVA, ORACLE TUNING PRODUCTS, AND A NEW PUZZLE.

Despite the fact that my column title changed in April, I am still traveling a lot. I went to Software Development '96 in San Francisco (March 25-29, 1996) to give a talk about SQL programming tricks. I had approximately 30 to 40 people in my session and I felt good about it when I finished because this is not the show at which I would normally draw a crowd. However, the real success at the show was the introductory Java session; it filled the original room and overflowed and filled a second room.

Java is an object-oriented language designed to run on a virtual stack machine that supports approximately 250 instructions. The idea is that a user signs on to the Internet and downloads a file of byte codes that the virtual Java machine executes locally. Old timers will remember this method being used for portable compilers developed by Nicholas Wirth. We called it pseudocode or p-code in those days and it also used a virtual stack machine.

First you wrote a program to simulate the p-code machine on your target computer. This took approximately a month. You then got Wirth's Pascal compiler as a file of p-code instructions and ran it through your own p-code machine program. This gave you a Pascal compiler on your target computer, which you could modify and optimize.

Java is a little different. You read a Java byte code file from the Internet, scan each instruction, and then execute it. Java has no pointers so there is no way to write a virus in the language or to overflow storage. That is very important for Internet work because you want some security when you load a program from an unknown source. There is talk of building a Java machine on a chip, so that you would install a circuit card on your machine and get an automatic hardware-level firewall.

Sun Microsystems released JDBC version 0.50, a Java SQL API specification, for public review. As I write this, the final version is scheduled for release in June 1996. JDBC defines a natural Java interface to the basic abstractions and concepts defined in the SQL/CLI. This document can be obtained from <http://splash.javasoft.com/jdbc/>.

Problems At NIST

Ken Gallagher turned in his resignation as WG3 DBL Rapporteur, effective after the May 1996 meetings of SC21 and WG3 in Kansas City, Missouri. This is due to a "reorganization and reprioritization of standardization

activities" by the National Institute of Standards and Technology (NIST). This is one of the dumbest things that a government agency could do. Your first clue is that the key sentence has three words that end in "-ization," which is always a sign of bureaucrats with time on their hands. But you really have to have been in government contracting work to know just how dumb this move is. I would guess that most of my readers have not followed the shift in the direction of NIST under the new administration. There is not much overlap between the readers of DBMS and the readers of Government Systems News. In a nutshell, the new director wants the labs to produce software and other products for the government instead of defining standards and test suites.

In the old days, before the Federal Information Processing Standards (FIPS), doing a federal procurement was a paper-work nightmare. Each piece of software and hardware had to be described and defined in painful detail, then related to the Request for Proposal specifications. You had something the size of a telephone book that cost tens of thousands of dollars to produce. All of this work and money was replaced with a single reference to the appropriate FIPS document.

NIST is still keeping standards and providing reference material for industry, education, and commerce. If you need to look up a physical constant, such as the speed of light in a vacuum, you can find it listed on a web site maintained by NIST: <http://physics.nist.gov/funcon.html>. This is the sort of thing that NIST should be doing, instead of developing software.

Oracle Tuning Products

Once in a while my beloved editor decides that the magazine needs to run a theme issue and he asks all of the columnists to do a piece on that theme. The rest of the time, we pretty much run amuck and only get edited for bad English and slander. This issue's theme is product reviews.

I have always liked query optimization as a research topic, so I thought I would look at third-party optimizers. Going back to basics, a SQL engine parses a query and produces an execution plan. This plan is usually kept in some sort of proprietary script language for display to the user when he requests it--DB2 uses the command EXPLAIN, Oracle uses ANALYZE, and so forth.

The query usually has several different possible execution plans, and the goal of the optimizer is to pick the best possible one, where "best" is usually defined as fastest performance. The trick is figuring out how to estimate the cost of a query from the test of the query and information in the schema. As a simple example, assume that you need to find all of the rows in a table in which ($x = 1$). If I have a table in which 99 percent of the rows meet the test condition, then using an index will waste time; but if only 5 percent of the rows meet the test condition, the index could help quite a bit. But then if the table is hashed or sorted on the x column, you might be better off jumping directly to the pages of physical storage that hold the rows in hash buckets or sequences. Oh yes, did I mention that you need to look at the rest of the query to see if there are other predicates hidden in referential constraints, CHECK() clauses, and VIEWS to combine into the search? The more you think about it, the more you realize that there is no such thing as a "simple query" when you try to optimize one.

The products I picked all work with Oracle. Bluntly, the three reasons

that Oracle is their target database are simple:

1. You cannot write a generic optimizer, so you must pick a target database.
2. Oracle has a 40 percent or better market share.
3. Oracle's own query optimization stinks.

Oracle finally included a cost-based optimizer in Oracle7. Before that release, Oracle performed a straight compilation on the SQL, just like a procedural language. The execution plan was totally dependent on how you wrote your SQL, down to the order of the tables in the FROM clause. Oracle's marketing people liked to call this "rules-based optimization" and tried to pass it off as a feature for years.

Platinum Plan Analyzer

Platinum Plan Analyzer (formerly Explain SQL from SQL Tools Inc., before being bought out in 1994) from Platinum Technology was developed by Edward Kosciuszko. His white papers and articles on query optimization are still available from Platinum and they are worth reading.

I do not have a great test laboratory to test a wide array of database products. In fact, I am still trying to get Windows 95 to run for more than three days straight on a brand new machine at home. This means that I have to rely on users.

Rich Holock of Pathfinder Solutions, a consulting firm, developed a private set of Oracle tricks for himself before he ran into Kosciuszko sitting in a booth passing out business cards at a trade show. Holock thinks that the strong point in Platinum's Plan Analyzer is that the product tracks index data distribution in its own tables, so it makes better decisions than the Oracle optimizer. Although this is not a fair question, I tortured Holock into estimating that Plan Analyzer beats the Oracle optimizer 70 percent of the time.

Holock also had a good "horror story" about clients who set up their execution plans when they first loaded data into a data warehouse and never bothered to update the execution plans as the database grew into the multi-gigabyte range. Think about how skewed the query execution plans must have become.

CA-ACE Insight

CA-ACE Insight from Computer Associates is a comprehensive Oracle SQL management tool. The product gives an impact report when you switch from rule-based to cost-based optimization or add an index to an existing table. There is a cross-reference report that locates similar SQL and PL/SQL statements so they can be converted into procedures.

Precise Software Solutions

Precise Software Solutions sells an Oracle tuning toolset under the name Precise/SQL, which is the parent of two other products in an integrated suite: Inspect/SQL and Analyze/SQL.

Inspect/SQL is an application-oriented monitor that differs from CA-ACE Insight and other database-oriented monitors. The typical scenario for

Inspect/SQL is to find the biggest user by the amount of resource usage and then drill down into data to find that user's worst SQL statement. The product has a Windows front end with a Unix agent, and a database agent to look at the network traffic. This lets you decide if the problem is on the network or the server by taking five to 10 samples per second. If the network is jammed with data, then the network needs work. If the network traffic is relatively light, then the server is not putting out data fast enough. Much as I hate to admit it, sometimes a performance problem requires a hardware solution.

Analyze/SQL is the next step in the process. It is much like Platinum's Plan Analyzer. Its unique feature is an "alternate SQL knowledgebase" that lets you find ways to rewrite the original SQL and then simulate the effect of those changes on the whole environment.

Pete Roberts, an information engineering specialist at Motorola Inc.'s Semiconductor Products Sector in Tempe, Arizona, was a beta user of Precise/SQL. He reported some cases in which queries went from two- to three-hour queries (and some that never came back at all) to average response times of 15 minutes or less.

You might want to ask Precise Software Solutions for a copy of its white paper "Application Tuning, the Missing Link" by Michael Abbey. (This is also available at <http://www.precisesoft.com/>.)

The Big Picture

The Big Picture from BitbyBit Software also analyzes all of an application's queries and the database as a whole, rather than just the individual queries. It works with Oracle and other DBMSs. The Big Picture extracts and analyzes all of the SQL embedded in a host language program. For example, it can modify or drop indexes based on their actual usage across the whole system, not just individual queries. I mention this because in the real world, programmers do not like to drop any indexes for fear that even though they don't use them, some other query might depend on that index. The result is that indexes accumulate like old padlock keys in a drawer.

The Big Picture can look for identical SQL in slightly different forms and let the user make them identical. This is very important in Oracle, because Oracle can share query and subquery results in cache among users without recomputing the queries, but only if the texts of the queries are identical.

The Big Picture also works with a wider range of SQL products (Oracle, DB2 family, Rdb, Sybase, SQL Server, XDB, Watcom, ODBC, and Informix). This lets you extract the SQL in an application and check if it will run against different back-end databases before you migrate your application.

Another SQL Puzzle

Larry Wade posted a version of this problem on the Microsoft Access CompuServe forum at the end of February 1996. He is running an employment service that has a database with tables for job orders, candidates, and their job skills. He is trying to perform queries to match candidates to job orders based on their skills. The job orders take the form of Boolean expressions connecting skills. For example, find all of the candidates with manufacturing and inventory or accounting skills.

First, let's construct a table of the candidates' skills. You can assume that personal information about the candidates is in another table, but I will not bother with it for this problem.

```
CREATE TABLE CandidateSkills
(candidateid INTEGER NOT NULL,
skill_code CHAR(15) NOT NULL,
PRIMARY KEY.(candidateid, skill_code));

INSERT INTO CandidateSkills VALUES (100, 'accounting');
INSERT INTO CandidateSkills VALUES (100, 'inventory');
INSERT INTO CandidateSkills VALUES (100, 'manufacturing');
INSERT INTO CandidateSkills VALUES (200, 'accounting');
INSERT INTO CandidateSkills VALUES (200, 'inventory');
INSERT INTO CandidateSkills VALUES (300, 'manufacturing');
INSERT INTO CandidateSkills VALUES (400, 'inventory');
INSERT INTO CandidateSkills VALUES (400, 'manufacturing');
INSERT INTO CandidateSkills VALUES (500, 'accounting');
INSERT INTO CandidateSkills VALUES (500, 'manufacturing');
```

The obvious solution would be to create dynamic SQL queries in a front-end product for each job order, such as:

```
SELECT C1.candidateid, 'jobid #212'--constant job id code
FROM CandidateSkills AS C1,--one correlation per skill
     CandidateSkills AS C2,
     CandidateSkills AS C3
WHERE C1.candidateid = C2.candidateid
     AND C1.candidateid = C3.candidateid
     AND
         --job order expression created here
         (C1.skill_code = 'manufacturing'
        AND C2.skill_code = 'inventory'
        OR C3.skill_code = 'accounting');
```

This statement is ugly and it will run forever, but it works. The order of evaluation of the Booleans in the final predicate will be handled by SQL's rules.

The working table will have the columns:

```
(C1.Candidateid, C2.Candidateid, C3.Candidateid, C1. skill_code,
C2.skill_code C3. skill_code)
```

and they will have all possible combinations of job skills. It will be huge:

```
(101, 101, 101, 'accounting', 'accounting', 'accounting')
(101, 101, 101, 'accounting', 'accounting', 'inventory')
(101, 101, 101, 'accounting', 'inventory', 'manufacturing')
(101, 101, 101, 'inventory', 'accounting', 'manufacturing')
(101, 101, 101, 'inventory', 'inventory', 'inventory')
(101, 101, 101, 'manufacturing', 'beet farming', 'shoemaking')
```

(101, 101, 101, 'manufacturing', 'inventory', 'accounting')

A good PowerBuilder or Delphi programmer can develop a screen form to do this in less than a week. You then save the query as a VIEW with the same name as the jobid code. Neat and quick! The trouble is that this solution will give you a huge collection of very slow queries. Got a better idea? See Puzzle Answer on page 24.

* BitbyBit Software, 119-A High St., Putney, London SW15 1SU, U.K.; 44-0-181-780-5300 or fax 44-0-181-780-5302; <http://www.netkonect.net/bitbybit/>; 100344.626@compuserve.com.

* Computer Associates International Inc., One Computer Associates Plaza, Islandia, NY 11788-7000; 800-225-5224, 516-342-5224, or fax 516-342-5734; <http://www.cai.com>; cainfo@cai.com.

* Platinum Technology Inc., 1815 South Meyers Rd., Oakbrook Terrace, IL 60181; 800-442-6861, 708-620-5000, or fax 708-691-0710; <http://www.platinum.com>; info@platinum.com.

* Precise Software Solutions Inc., 50 Braintree Hill Park, Ste. 110, Braintree MA 02184; 800-310-4777, 617-380-3300, or fax 617-380-3349; <http://www.precisesoft.com/>; 75703.117@compuserve.com.

Joe Celko is a member of the ANSI X3H2 Database Standards Committee, a widely published author, and a consultant with OSoft Development Corp. in Atlanta. Joe is also a frequent contributor to the DBMS Forum on CompuServe. You can email Joe at 71062.1056@compuserve.com.

Type: Column

Record#: 18 456 807

*** End ***